1. In Google Docs, create 5 test cases for functional testing of the [https://www.carsforsale.com](https://www.carsforsale.com/) website (5 top priority tests).   
   Link the document in LMS.

**ANSWER**

* Test the sign in functionality in the header to be sure that users can create account and sign in successfully
* Test search dropdown functionality in the header to be sure users can access all its features
* Test “FIND YOURS” functionality and all it features
* Verify that users can search for cars by brand name by clicking on the brand names on the home page
* Verify that users can message car sellers by email, call and text message.

1. **With** [**https://reqbin.com/**](https://reqbin.com/)**, test the following API** [**Reqres**](https://reqres.in/) **methods:**
   * **LIST USERS**
   * **SINGLE USER**
   * **CREATE**
   * **UPDATE (PUT)**
   * **UPDATE (PATCH)**
   * **DELETE**
   * **REGISTER - SUCCESSFUL**
   * **LOGIN - SUCCESSFUL**

**Create a document and describe in it the bugs you found. The description can be of any format, but must include screenshots. Add the document to LMS.**

* + **LIST USERS**

|  |  |
| --- | --- |
| **Summary** | **Get the List of users from database** |
| **Project** | [**https://reqres.in/api/users**](https://reqres.in/api/users) |
| **method** | GET |
| **Status Code** | 200 (ok) |
| **Time and size** | 205 ms, 0.97 kb |
| **Response (content)** | {  "page": 1,  "per\_page": 6,  "total": 12,  "total\_pages": 2,  "data": [{  "id": 1,  "email": "george.bluth@reqres.in",  "first\_name": "George",  "last\_name": "Bluth",  "avatar": "https://reqres.in/img/faces/1-image.jpg"  }, {  "id": 2,  "email": "janet.weaver@reqres.in",  "first\_name": "Janet",  "last\_name": "Weaver",  "avatar": "https://reqres.in/img/faces/2-image.jpg"  }, {  "id": 3,  "email": "emma.wong@reqres.in",  "first\_name": "Emma",  "last\_name": "Wong",  "avatar": "https://reqres.in/img/faces/3-image.jpg"  }, {  "id": 4,  "email": "eve.holt@reqres.in",  "first\_name": "Eve",  "last\_name": "Holt",  "avatar": "https://reqres.in/img/faces/4-image.jpg"  }, {  "id": 5,  "email": "charles.morris@reqres.in",  "first\_name": "Charles",  "last\_name": "Morris",  "avatar": "https://reqres.in/img/faces/5-image.jpg"  }, {  "id": 6,  "email": "tracey.ramos@reqres.in",  "first\_name": "Tracey",  "last\_name": "Ramos",  "avatar": "https://reqres.in/img/faces/6-image.jpg"  }],  "support": {  "url": "https://reqres.in/#support-heading",  "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  }  } |

* + **SINGLE USER**

|  |  |
| --- | --- |
| **Summary** | **Get a Single user from the database.** |
| **Project** | [**https://reqres.in/api/users/1**](https://reqres.in/api/users/1) |
| **method** | GET |
| **Status Code** | 200 (ok) |
| **Time and size** | 23 ms, :0.27 kb |
| **Response (content)** | {  "data": {  "id": 1,  "email": "george.bluth@reqres.in",  "first\_name": "George",  "last\_name": "Bluth",  "avatar": "https://reqres.in/img/faces/1-image.jpg"  },  "support": {  "url": "https://reqres.in/#support-heading",  "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  }  } |

* + **CREATE**

|  |  |
| --- | --- |
| **Summary** | **CREATE a page of users in the database** |
| **Project** | [**https://reqres.in/api/users**](https://reqres.in/api/users/1)?page=3 |
| **method** | POST |
| **Status Code** | 201 |
| **Time and size** | 198 ms, 1.04 kb |
| **Response (content)** |  |

* + **UPDATE (PUT)**

|  |  |
| --- | --- |
| **Summary** | **update a user with “ Id 2” in the database** |
| **Project** | [**https://reqres.in/api/users**](https://reqres.in/api/users/1)/2 |
| **method** | PUT |
| **Response (content)** |  |

* + **UPDATE (PATCH)**

|  |  |
| --- | --- |
| **Summary** | **update only the user email with “ Id 2” in the database** |
| **Project** | [**https://reqres.in/api/users**](https://reqres.in/api/users/1)/2 |
| **method** | PATCH |
| **Response (content)** |  |

* + **DELETE**

|  |  |
| --- | --- |
| **Summary** | **Delete only the user with “ Id 2” from the database** |
| **Project** | [**https://reqres.in/api/users**](https://reqres.in/api/users/1)/2 |
| **method** | DELETE |
| **Response (content)** |  |

* **REGISTER - SUCCESSFUL**

|  |  |
| --- | --- |
| **Summary** | **Register user on the database** |
| **Project** | [**https://reqres.in/api/users**](https://reqres.in/api/users/1)/register |
| **method** | POST |
| **Response** |  |

* **LOGIN - SUCCESSFUL**

|  |  |
| --- | --- |
| **Summary** | **Login user on the database** |
| **Project** | [**https://reqres.in/api/users**](https://reqres.in/api/users/1)/login |
| **method** | POST |
| **Response** |  |

**BUGS**

|  |  |
| --- | --- |
| Unregistered users are able to login |  |
| Create users with already existing Id and other user details |  |
| Allows user registration with other information than email and password |  |
| Create but does not save created users content in database. For example the user below has been created in database, but upon requesting the same user with a GET request it returns error  {  "email": "tobiasfunke.funke@reqres.in",  "first\_name": "Tobioluwa",  "last\_name": "Funkeline",  "id": "627",  "createdAt": "2023-08-16T12:01:52.203Z"  } |  |
| Returns ok and Create empty data page for a GET request when user request for pages that do not exist  For example requesting for page 3 which does not exist; <https://reqres.in/api/users/?page=3> |  |

1. Use <https://reqbin.com/> to send one valid request for each method to [GO REST](https://gorest.co.in/) (**/users** endpoint).
   * After sending each request, compare its details entered in <https://reqbin.com/> and displayed in Developer Tools.

Create a document with screenshots showing the correspondence between the data in both tools. Add the document to LMS.

|  |  |  |
| --- | --- | --- |
| Data | Redbin.com | Developer tool |
| Responses for a POST request  <https://gorest.co.in/public/v2/users> |  |  |
|  |  | This request has no response data available |
| GET request: <https://gorest.co.in/public/v2/users> |  |  |
|  |  |  |
| PUT request |  |  |
|  |  |  |
| PATCH request |  |  |
| DELETE request |  |  |